

High Data Rate Architecture (HiDRA)

Alan Hylton
alan.g.hylton@nasa.gov *

Dr. Daniel Raible
daniel.e.raible@nasa.gov †

One of the greatest challenges in developing new space technology is in navigating the transition from ground based laboratory demonstration at Technology Readiness Level 6 (TRL-6) to conducting a prototype demonstration in space (TRL-7). This challenge is compounded by the relatively low availability of new spacecraft missions when compared with aeronautical craft to bridge this gap, leading to the general adoption of a low-risk stance by mission management to accept new, unproven technologies into the system. Also in consideration of risk, the limited selection and availability of proven space-grade components imparts a severe limitation on achieving high performance systems by current terrestrial technology standards. Finally from a space communications point of view the long duration characteristic of most missions imparts a major constraint on the entire space and ground network architecture, since any new technologies introduced into the system would have to be compliant with the duration of the currently deployed operational technologies, and in some cases may be limited by surrounding legacy capabilities. Beyond ensuring that the new technology is verified to function correctly and validated to meet the needs of the end users the formidable challenge then grows to additionally include: carefully timing the maturity path of the new technology to coincide with a feasible and accepting future mission so it flies before its relevancy has passed, utilizing a limited catalog of available components to their maximum potential to create meaningful and unprecedented new capabilities, designing and ensuring interoperability with aging space and ground infrastructures while simultaneously providing a growth path to the future.

The International Space Station (ISS) is approaching 20 years of age. To keep the ISS relevant, technology upgrades are continuously taking place. Regarding communications, the state-of-the-art communication system upgrades underway include high-rate laser terminals. These must interface with the existing, aging data infrastructure. The High Data Rate Architecture (HiDRA) project is designed to provide networked store, carry, and forward capability to optimize data flow through both the existing radio frequency (RF) and new laser communications terminal. The networking capability is realized through the Delay Tolerant Networking (DTN) protocol, and is used for scheduling data movement as well as optimizing the performance of existing RF channels. HiDRA is realized as a distributed FPGA memory and interface controller that is itself controlled by a local computer running DTN software. Thus HiDRA is applicable to other arenas seeking to employ next-generation communications technologies, e.g. deep space. In this paper, we describe HiDRA and its far-reaching research implications.

*Communications and Intelligent Systems Division, NASA GRC, Cleveland OH

†Communications and Intelligent Systems Division, NASA GRC, Cleveland OH

I. Introduction

*“Indeed, we have the know-how, but we do not have the know-why,
nor the know-what-for.”* -Erich Fromm

The High Data-Rate Architecture (HiDRA) project was established at NASA Glenn in order to address rate asymmetries in space data and communications systems. Historically, communications has placed a considerable constraint on data return, and now as the physical layers improve the rest of the system must be matched to take full advantage of such gains. HiDRA is an architectural study designed to create an evolvable platform that meets near-term requirements, e.g. those of the Integrated Radio and Optical Communications (iROC) project, with a concurrent goal of enabling 100+Gbps communications. The HiDRA study is based on software and hardware driven by Delay Tolerant Networking (DTN) in order to optimize data flow and return while respecting CCSDS standards. Moreover, DTN enables the agglutination of HiDRA-equipped nodes into an otherwise heterogeneous managed and secure network.

The bandwidth of scientific space instruments has easily outpaced space communications systems. There have been notable improvements in the physical layer; see, e.g., the Lunar Laser Communications Demo (LLCD) [7]. While LLCD was successful in its mission to demonstrate laser communications from the moon to Earth, it also demonstrated how internal rate mismatches degrade performance. LLCD could transmit data at 622Mbps, but its connection to its host, the Lunar Atmosphere and Dust Environment Explorer (LADEE), was constrained to 40Mbps. When LADEE needed to dump its gigabyte of memory, it would have taken every RF pass for three days. This was not acceptable, as LADEE was designed to spiral into the moon - time was of the essence. As LLCD had proven itself prior to this need, they used it, and it took roughly 4 minutes. At full-rate, LLCD could have sent on the order of 19 gigabytes, but instead most of its data return was pseudorandom binary sequence (PRBS) data. This is to be expected of a demo, yet is indicative of a common problem. There are gaps in research and development of various layers in the communications stack, and systems of systems might be of varying vintages, capabilities, and constraints.

Bridging the gap between a ground-based technology research and development effort and a space-based technology demonstration is a formidable challenge. This challenge is compounded by the relatively low availability of new spacecraft missions when compared with aeronautical craft to bridge this gap, leading to the general adoption of a low-risk stance by mission management to accept new, unproven technologies into the system. Also in consideration of risk, the limited selection and availability of proven space-grade components imparts a severe limitation on achieving high performance systems by current terrestrial technology standards. Finally from a space communications point of view the long duration characteristic of most missions imparts a major constraint on the entire space and ground network architecture, since any new technologies introduced into the system would have to be compliant with the duration of the currently deployed operational technologies, and in some cases may be limited by surrounding legacy capabilities. Beyond ensuring that the new technology is verified to function correctly and validated to meet the needs of the end users the formidable challenge then grows to additionally include: carefully timing the maturity path of the new technology to coincide with a feasible and accepting future mission so it flies before its relevancy has passed, utilizing a limited catalog of available components to their maximum potential to create meaningful and unprecedented new capabilities, designing, and ensuring interoperability with aging space and ground infrastructures while simultaneously providing a growth path to the future.

There is no better case study of this than with the International Space Station (ISS) – one of the crowning accomplishments of humankind, an engineering marvel, but at around 20 years old it also may be described as aging infrastructure. The last several decades have seen a focus on physical (PHY) layer technology developments in aperture and amplifier technologies to support such systems as higher frequency radio frequency (RF) and even laser communications terminals. These state of the art communication system upgrades underway will help to realize higher data rates, but must interface with the existing data infrastructure. As new communication terminals are presented to the ISS, a considerable rate mismatch will be introduced between the existing and emerging capabilities. The HiDRA project is designed to provide networked store, carry, and forward capability to optimize data flow through the system. The networking capability is realized through the DTN protocol, and is used for scheduling data movement as well as optimizing the performance of existing RF communications channels. Moreover, HiDRA is designed to be as transparent as possible

while simultaneously supporting Multiple-In Multiple-Out (MIMO) operations between optimizing the performance within the purview of the spacecraft data bus and the multiple communication channels over the free-space network. HiDRA is realized as a distributed FPGA memory and interface controller that is itself controlled by a local computer running DTN software. One may visualize HiDRA as a water tower of data that is controlled by networking software. One may also consider HiDRA to be, at least locally, a hub in a hub-and-spoke style network. Thus HiDRA is applicable to other arenas seeking to employ next generation communications technologies, e.g. deep space.

II. Defining the Problem

The introduction of emerging physical-layer technologies into the space communication networks is creating a heterogeneous system between new and old, high rate and low rate, optical and RF. Early research has examined network management of dissimilar RF and optical link architectures in the near-Earth environment [5, 11], and have taken advantage of the short time-of-flight characteristics to create workable network solutions. Unfortunately many of the techniques and parameter tunings are not extensible to the deep space domain due to the inherent dynamic differences between the environments and the lack for real time feedback to control from. A multi-hop multi-path hybrid RF and optical test bed has been constructed to emulate future networks and to support protocol and hardware refinement utilizing the ION implementation of DTN [6]. Initial results characterized several challenges and evaluated the effectiveness of DTN as a solution to mitigate them, revealing the need for significant amounts of local high speed memory to accommodate large and numerous bundles sent across high data rate physical layers. Further challenges associated with the Bundle Protocol Specification include the lack of reliability checks within the DTN bundle (the fundamental unit of data in a DTN), varying support for fragmentation, lack of definition for convergence layers, a flat address space creating difficulty in scaling and routing, and no standardized discovery mechanism [2].

Adoption of DTN into future high speed space networks, and especially those realized by laser communications, hinges on the ability to successfully transmit data in at least the Gb/s order of magnitude range. A successful test was performed at JPL with the ION implementation running over a Free-Space Optical (FSO) network [12]. Forcing the central processing unit (CPU) to move data from non-volatile storage to RAM to the communications system interface at these rates would cause undue burden and bottlenecking. A potential solution being researched is the partial implementation of ION, JPL's implementation of DTN, in FPGAs to affect a form of direct memory access (DMA). Offloading the non-computational overhead functionality over to hardware implementation will streamline the data flow and separate it from the overhead sequential processing. This will decrease ION's footprint without adding excessive complexity to the rest of the system. In order to maintain flexibility and the ability to update the protocol, most of ION would remain in software form on the CPU. Early experiments of this paradigm have examined the implications of custody transfer on the distribution of transfers and the inclusion of Contact Graph Routing (CGR) to allow establishment of one link to preclude all others – at least when they share a common outduct [9].

HiDRA is envisioned as a sort of glorified hard drive controller. We have storage that is controlled by a computer. It will mostly collect data through the computer that directly controls it, and will send that data out a specified port in a specified manner. The novelty comes in defining the demarcation between hardware, software, and DTN. Certain parts of DTN clearly belong in software, e.g. routing. Data flow is best realized in hardware, and if possible, beyond the bus arbitration of the main computer. We realize this by connecting FPGAs to computers via PCI Express. These FPGAs then have dedicated storage (volatile or non-volatile) and may be connected to a variety of radios or networks. It is less clear how HiDRA would best interact with the software, and in particular, the networking component. If we add the overhead of making our memory into a hard drive with the capability to select the paths that the data takes, particularly using a standard file system, we will add complexity in the drivers and in the hardware, but will likely have less work when integrating DTN. If we use HiDRA as a simple memory controller, where we consider the storage is treated as RAM, then the hardware and drivers are simple, but the DTN integration is less straightforward. ION features its own memory allocation algorithms for working within its heap. It is possible to extend these to HiDRA. Moreover, ION is a non-monolithic DTN implementation where the various components interprocess communication is achieved via reading and writing common memory. It is then possible to write

a program that communicates with ION and replaces its bundle storage capability with a HiDRA-specific algorithm. Revisiting the file system notion, we could also use a more complicated controller to simplify bundle storage and processing. The key will be to research these trades to find out how retransmission is best achieved.

Retransmission happens at multiple layers. We might retransmit an entire bundle if need be, or if we use a lower-layer protocol that features reliability, such as the Licklider Transmission Protocol (LTP), we may need to store all or parts of a bundle until we have either received an acknowledgment or have given up. We have discussed HiDRA as hardware, but originally as an architecture. Reliability means different things in different arenas. If we have our water tower of data in Low-Earth Orbit (LEO), essentially real time feedback is possible and encouraged; we can use ACK and NACK based protocols depending on forward link considerations. The link can reactively fragment to current conditions. However, HiDRA is also considered for deep space. iROC is a Mars-to-Earth laser communications project that utilizes a co-located, co-boresighted telescope and antenna (teletenna) to provide both RF and optical transmission. The is automatically a multi-path network, and as the optical and RF ground stations are multiple and not spatially co-located, the network is also multi-hop. Moreover, iROC is designed to relay data, thus adding to the network complexity. This complexity is both in terms of the network topology, as iROC is not a leaf node, but also in terms of data handling requirements. Indeed, as new projects come and go, with updates in technology, the utility of iROC will depend on its buffer size and policies, among other challenges. From Mars we cannot employ a network model that relies on real time feedback. Fragmentation and link adjustments must be made proactively, and the time required to get acknowledgments (which may happen at the bundle layer) will increase. This plays a heavy hand in driving the buffer sizing requirement.

Further, we consider a wide range of data rates. The target goal is to make and demonstrate an instance of HiDRA that can transmit data at 100Gbps or beyond. This must use cutting-edge hardware that is not necessarily qualified for space operations. For iROC, we must transmit data up to the low Gbps range using radiation hardened hardware. We then turn to the TCP Offload Engine (TOE) for an example. The TOE handles TCP/IP overhead (up to and including handshaking). In [3], we see that the 1GHz/Gbps rule of TCP/IP breaks down only with modern, terrestrial CPU speeds, and this relies on how data is aggregated. Given limitations of deep space processing, offloading the largely non-computational burden of bundle overhead bears merit, and even in LEO as we strive for 100Gbps and beyond, we will rely on hardware acceleration. This not only makes DTN and networking in space viable, but it also provides fertile ground for DTN optimization research once DTN is more operational and usage data have been collected.

Thus the hardware particulars of HiDRA instantiations will be partially application dependent, however these variations will not influence how DTN, the computer, and the FPGA communicate, nor the boundary between them. Therefore we strive to find and develop the common ground. The essence of HiDRA is, then, to develop a networked buffering solution that is general enough to fit a host of situations without being so general as to be unrealizable.

III. Building Towards Specific Use-Cases

We will consider the deep-space scenario with a look towards a near-earth demonstration.

III.A. iROC

We began developing HiDRA considering the deep space use-case. iROC will most likely use previous-generation Virtex 5 FPGAs as radiation hardened versions exist. iROC is illustrated in Figure 1. While there are many technologies being developed to make iROC possible, we will focus on the networking implications. In particular, communications from Mars to iROC might occur over the teletenna or over other RF receivers. If the teletenna is used, this precludes iROC from communicating to Earth. Depending on where Mars, the Earth, the Sun, and iROC are, said communication might be impossible regardless of spacecraft attitude. Given power constraints, either RF or optical will be used, but not simultaneously. RF would broadcast at a slower rate whereas optical would unicast at a quicker rate. Finally, depending on where the optical ground stations are ultimately built, terrestrial infrastructure must be built. Network management, and in particular

policies, will be required to ensure that iROC can not only service multiple missions but also remain relevant as missions might be added or removed. Finally, if security in the form of encryption becomes desired in deep space applications, it is worth exploring any unused portion of the FPGA to offload computationally expensive encryption from the CPU.

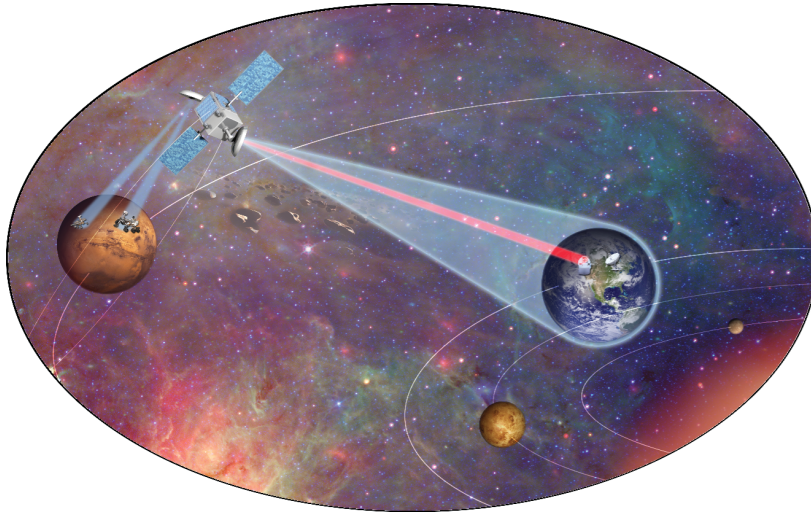


Figure 1: iROC Conceptual Illustration

mode/two-step approach offers an incremental increase in complexity going from a starting base utilizing simplified Ethernet interface to final mode utilizing SFP interface. Also, the two mode approach allows the SFP interface to be purchased/developed/debugged in parallel with the first mode (X-ES System → Ethernet Receiver).

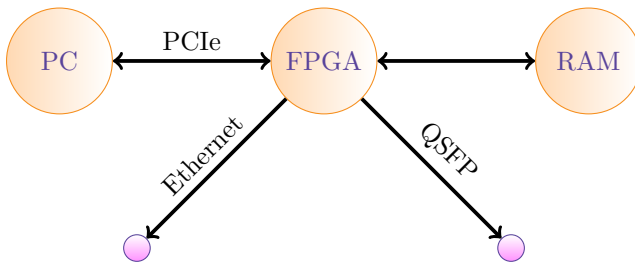


Figure 2: X-ES System

The X-ES box will be solely a transmitter, as in Figure 2. The data to be sent will be buffered in the FPGA card’s dedicated RAM. The first step is to have the X-ES box transmit via Ethernet, or colloquially, have Figure 2 talk to Figure 3. The follow-on step is to then add QSFP connectivity; that is, have Figure 2 talk to Figure 4. The RAM in the diagram is the FPGA’s dedicated, on-board memory, which shall be used in both cases. The FPGA shall communicate to the PC via PCIe, and the drivers will be interrupt driven.

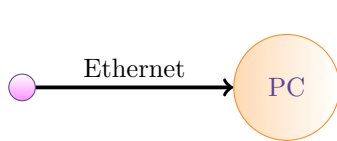


Figure 3: Ethernet Receiver

The Ethernet receiver is just a PC with a NIC, as shown in Figure 3. The PC will listen for UDP packets, and software will process them as they are received. This will get us started, help us debug and develop the drivers for the X-ES box, and can later be used as we stress the multiple out in the MIMO design.

The SFP receiver will be a PC with an VC709 development board, as illustrated in Figure 4. The RAM in the diagram is the VC709’s on-board memory. There will be a buffer in the memory that the recipient FPGA writes incoming data to. Our testbed uses the AlphaData FPGA in the X-ES box as the transmitter (i.e., the iROC satellite). Using its QFSP cage, we connect it to the VC709 FPGA development board in the SFP receiver. The purpose is to have a straightforward means by which we can integrate the hardware with the optics lab of iROC and other projects.

As shown, all three of these systems exist. We store, carry, and forward data upon separate commands, thus demonstrating DTN-like functionality. The Ethernet system functions at 1Gbps, and the SFP system transmits at 1.4Gbps. In software, drivers have been written so that the storage appears as a blank slate of memory. Therefore, while we are encroaching on the boundary between hardware and DTN, we can only now beginning to probe it in earnest now that the testbed is operational. We include three notional (and simple!) process diagrams: the Ethernet receiver process (Figure 5), the SFP receiver process (Figure 6), and the X-ES transmitter process (Figure 7). When reading these diagrams, we consider time to be increasing as we move down the page. The active members of the diagrams are represented by vertical lines with their labels both at the top and at the bottom. Interaction between them is represented by arrows. The placement of the arrows is not meant to suggest that there is no parallelism. The simplest process to start with is the Ethernet receive process, which is suggested in Figure 5. The left-hand vertical line, labeled “X-ES,” then, refers to the entire X-ES system, FPGA and all (See Figure 2). On the right, the PC refers solely to the otherwise disjoint PC that is the Ethernet receiver (See Figure 3).

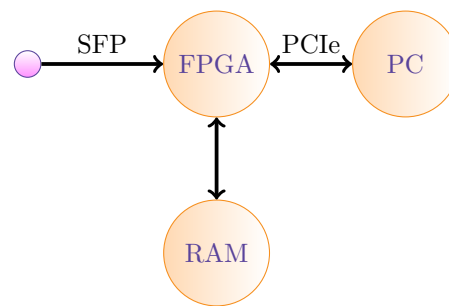


Figure 4: SFP Receiver

The SFP receiver process is the next step up in complexity, and is shown in Figure 6. Here the X-ES box is considered a self-contained unit, which means the X-ES PC, X-ES FPGA, and X-ES FPGA memory. The FPGA, RAM, and PC specified in Figure 6 refer to SFP receiver, and thus the FPGA and RAM are the VC709 board. Many details are obviously omitted, such as how the data gets from the RAM to the FPGA. Again, the arrows are not meant to imply that there is no parallelization. The X-ES transmit process is suggested in Figure 7. Here we consider this diagram to be of the internals of the X-ES box as a whole, so the FPGA refers to the AlphaData card, and the RAM is the AlphaData card’s on-board memory. The Ethernet and QSFP refers to the physical out-duct. It has been suggested that there are blocking and non-blocking calls for DMA transfer, and as such, we can decide if we need interrupts to know when sending data from the X-ES PC to the X-ES FPGA’s DDR for storage. The SFP system communicates using the Aurora protocol, and as suggested by the diagrams, all current communication is unidirectional.

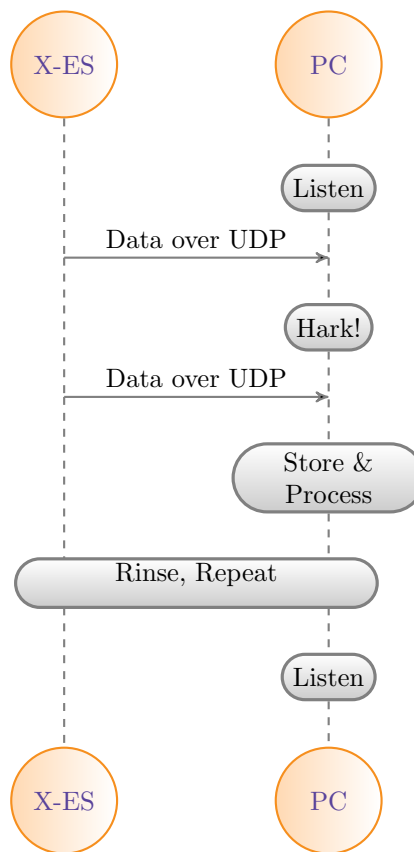


Figure 5: Ethernet Receive Process

We have been able to stream video and reliably transmit data (up to 2GB, the amount of on-board storage of the X-ES box) using the Ethernet model. This is fully configurable in software, i.e., we set the IP address, the data rate, and so forth using a driver call which sets registers within the FPGA, implying that no further hardware configuration is required. This immediately allows for flexibility in the testbed.

After base DTN integration has been established, communications will be made bidirectional to support

iROC as a relay satellite. It should be noted that orbital analyses have been conducted and as such contact schedules can be realistically created. In particular, there is no current competition for time on a deep-space optical receiver, granting flexibility in scheduling. Therefore connectivity models are available, and indeed have been used for previous DTN testing. Traffic models are also considered.

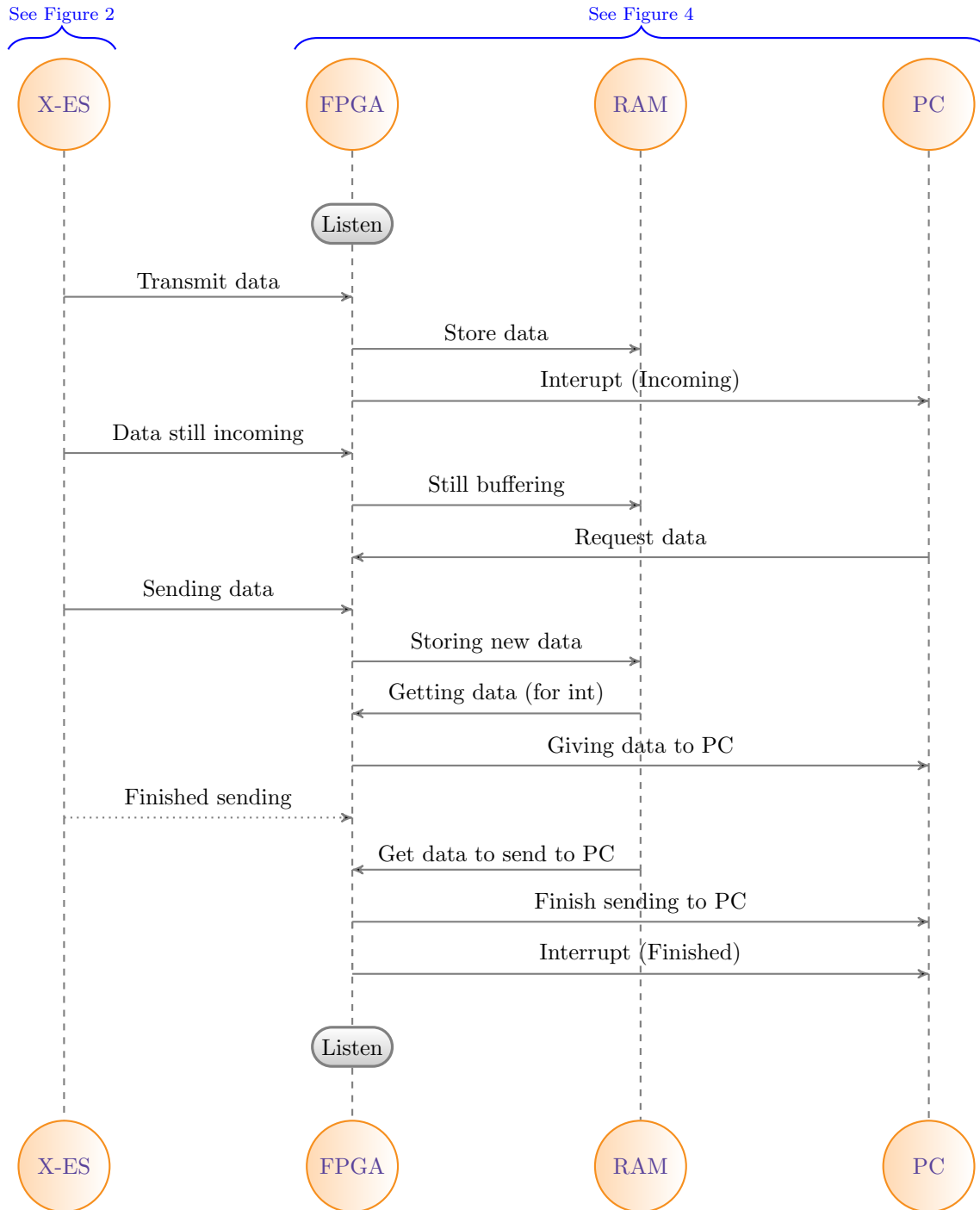


Figure 6: SFP Receive Process

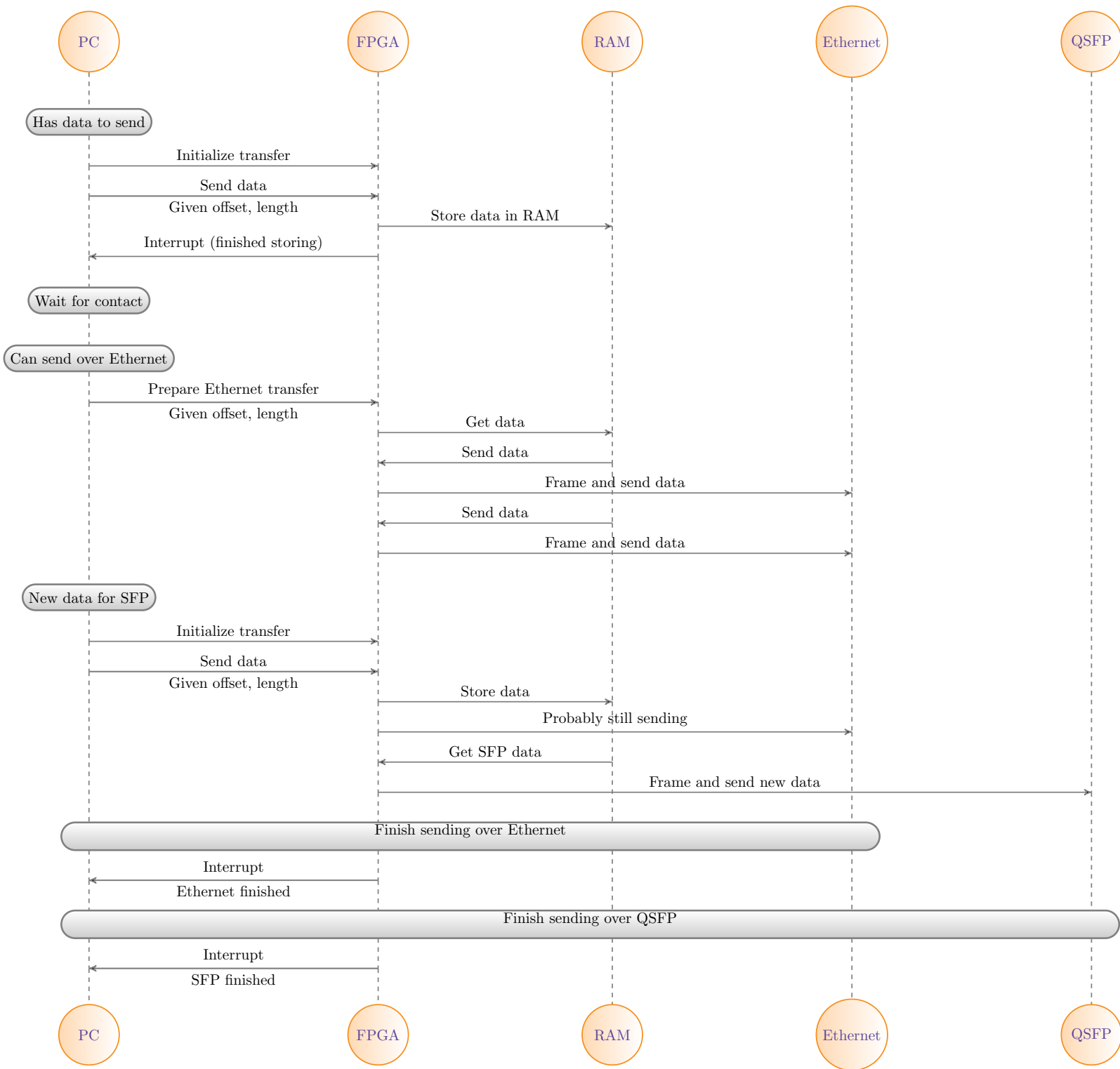


Figure 7: X-ES Transmit Process

III.B. ISS

The ECOSystem Spaceborne Thermal Radiometer Experiment on Space Station (ECOSTRESS) project will measure plant transpiration from the ISS from 2017-2019 [10]. ECOSTRESS will collect data at an average of 2.6Mbps, and can operate continuously, yielding an average of 28 gigabytes of a data per day. At peak performance, it can yield up to 47 gigabytes.



Figure 8: ECOSTRESS planned coverage. Note instrument is capable of additional coverage and can be commanded to acquire data over additional regions given sufficient downlink. (Taken directly from [8])

relay terminal in geosynchronous orbit. LCRD will support uncoded data from 72Mbps to 2.88Gbps [4]. Presuming that we use a code rate of $1/2$, we could transmit all 47GB of data in 261.1 seconds, or under 4.5 minutes. Given future optical upgrades to ISS, per pass a 5 minute span from ISS to LCRD is definitely achievable, and up to 25 minutes is possible. As there are roughly 16 passes per day, by using HiDRA to buffer data from ECOSTRESS we can easily meet current needs while providing plenty of headroom to support future science developments.

In addition to our iROC use-case development, we are concurrently developing an as of 2016 cutting-edge platform using Xilinx Virtex Ultrascale FPGAs. We are using the Bittware XUSP3R development board which feature a Xilinx Virtex UltraScale 190, 256GB of DDR4 RAM, a PCIe interface, and four QSFP28 cages. This development board supports up to 400Gbps communications [1]. The ECOSTRESS project will communicate using the DTN protocol on a laptop on-board the ISS. We will provide an interface between the laptop and the upcoming optical upgrades; the interface to the laptop will be PCIe (via PCIe breakout boxes that connect to laptops via ExpressCard 54, which supports speeds up to 5000 Mbps). The interface to the laser terminal will be Aurora over a physical connection to be specified at a later date.

By forging a path from the science payloads to upcoming optical terminals, which will necessarily encounter disruption and hand overs, we can greatly loosen the global constraints that communications have historically placed on data return. Locally, we introduce a missing component needed to realize space networking.

IV. Conclusion and Future Research

HiDRA as a network-managed buffer has clear utility for tying systems together, and while even a rudimentary instance would prove useful, it gives rise to many research projects.

As mentioned, there are several means by which DTN might utilize HiDRA, and there is a balance to be struck between software and hardware. As we move from hundreds of Mbps to low-Gbps rates, any functional style will likely suffice. The ultimate goal of 100Gbps and beyond, however, might require further modification. Consider the probably use-case of such a link. Most probably, it would be a highly bursty link with contact times ranging from seconds to minutes. If routing computations and other signaling create a latency of n seconds, then n Gbps of data were not transmitted. Thus there is an effort to mitigate this. So far, we synchronize an internal clock in the FPGA with the host computer and plan to use DTN to preemptively command the FPGA, say 30 seconds ahead of a known contact. Other optimizations might suffice, but will depend on how deterministic this latency is.

In addition to non-HiDRA specific DTN research, buffering requirements must be researched. This includes extrapolating for future missions. Particularly, if precedent is set for 1.44Gbps communications from ISS, we can expect a rapid increase in future project requirements. This also forces issues regarding network management and policies.

Further consideration of ISS future-proofing, to the extent possible, involves creating a custom board adding a variety of connectors. But another avenue would be considering multiple optical links on different physical ends of the ISS; the infrastructure would be the bottleneck, implying that multiple HiDRAs would be needed for concurrent operation. Load Optimizing load balancing introduces a door for cognitive networking.

Despite starting relatively recently, the HiDRA team has made progress towards the boundaries between software, hardware, and DTN. Store, carry, and forward has been demonstrated, configurability via drivers has been demonstrated, and video streaming has been demonstrated. Research into queuing and data management as well as DTN are the next hurdles.

V. Acknowledgments

The authors would like to thank the NASA Space Communications and Navigation (SCaN) program, and in particular Dr. Don Cornwell for supporting this research, and Mr. David Chelmins for his support.

References

- ¹BittWare.com/Xilinx — XUSP3R, 2016. [Online; accessed August-2016].
- ²A. Hylton, D. Raible. Networked Operations of Hybrid Radio Optical Communications Satellites. *32nd AIAA International Communications Satellite Systems Conference*, 2014.
- ³Annie P. Foong, Thomas R. Huff, Herbert H. Hum, Jaidev P. Patwardhan, Greg J. Regnier. TCP Performance Re-Visited. 2003.
- ⁴Bernard L. Edwards, Dave Israel, Keith Wilson, John Moores, Andrew Fletcher. Overview of the Laser Communications Relay Demonstration Project. *SpaceOps2012*, 2012.
- ⁵Vincent W. S. Chan. Optical Satellite Networks. *Journal of Lightwave Technology*, VOL. 21, NO. 11, November 2003.
- ⁶D. Raible, A. Hylton. Integrated RF/Optical Interplanetary Networking Preliminary Explorations and Empirical Results. *30th AIAA International Communications Satellite Systems Conference*, 2012.
- ⁷Don M. Boroson, Bryan S. Robinson, Daniel V. Murphy, Dennis A. Burianek, Farzana Khatri, et al. Overview and results of the Lunar Laser Communication Demonstration. *Proc. SPIE 8971, Free-Space Laser Communication and Atmospheric Propagation XXVI, 89710S*, March 6, 2014.
- ⁸Simon Hook. Science - ECOSTRESS, 2016. [Online; accessed August-2016].
- ⁹Hylton, D. Raible, J. Juergens, D. Iannica. On Applications of Disruption Tolerant Networking to Optical Networking in Space. *30th AIAA International Communications Satellite Systems Conference*, 2012.
- ¹⁰Julie A. Robinson, William L. Stefanov. Earth Science Research on the International Space Station. *Committee on Earth Science and Applications from Space (CESAS) Space Studies Board National Academies of Science, Engineering, Medicine*, 2016.
- ¹¹Pam Clark, Arjan Sengers. Wireless Optical Networking Challenges and Solutions. *Military Communications Conference*.
- ¹²Schoolcraft, J.; Wilson, K. Experimental characterization of space optical communications with disruption-tolerant network protocols. *Space Optical Systems and Applications*, 2011.